Nonparametric Monotone Classification with MOCA

Nicola Barile
Universiteit Utrecht
barile@cs.uu.nl

Ad Feelders
Universiteit Utrecht
ad@cs.uu.nl

Abstract

We describe a monotone classification algorithm called MOCA *that attempts to minimize the mean absolute prediction error for classification problems with ordered class labels. We first find a monotone classifier with minimum L_1 loss on the training sample, and then use a simple interpolation scheme to predict the class labels for attribute vectors not present in the training data. We compare* MOCA *to the Ordinal Stochastic Dominance Learner (*OSDL*), on artificial as well as real data sets. We show that* MOCA *often outperforms* OSDL *with respect to mean absolute prediction error.*

1 Introduction

Monotonicity constraints occur frequently in data mining problems and such constraints can be elicited from subject area experts with relative ease and reliability. This has motivated the development of data mining algorithms that are able to enforce such constraints in a justified manner.

In this paper we present MOCA, an algorithm for nonparametric monotone classification in problems with ordered class labels. The algorithm consists of two basic components. First, a monotone classifier is built that minimizes L_1 loss on the training data. This classifier is only defined on the observed input vectors. To extend it to the complete input space, a straightforward interpolation scheme is used that is guaranteed to preserve the monotonicity property. To determine the class allocation for a given input vector, MOCA estimates the class distribution for that input vector, and then assigns it to the (smallest) median class. Estimation of the class probability distribution is performed in such a way that allocation to the median satisfies the monotonicity property.

The paper is organized as follows. In the next section, we establish some notation and definitions that are used throughout the paper. In section 3 we discuss isotonic regression, a technique that is essential to MOCA. In section 4 we discuss how MOCA estimates the class probability distributions, and the MOCA allocation rule. We show that the allocation rule minimizes L_1 loss on the training data. Furthermore, we show how new data points are predicted with a straightforward interpolation scheme. In section 5 we discuss related work, in particular OSDL, a system that is intended for similar problems as MOCA. After that, we illustrate MOCA and OSDL through a small example in section 6. In section 7 we perform an experimental comparison of OSDL and MOCA on both artificial and real data sets. Finally, we draw conclusions in section 8.

2 Preliminaries

Let $\mathbf{X}$ denote the vector of predictors (attributes), which takes values $\mathbf{x}$ in a p-dimensional input space $\mathcal{X} = \times \mathcal{X}_i$, and let Y denote the class variable which takes values y in a one-dimensional space $\mathcal{Y} = \{1, 2, \ldots, k\}$, where k is the number of class labels. Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote the set of observed data points in $\mathcal{X} \times \mathcal{Y}$, and let Z denote the set of distinct $\mathbf{x}$ values occurring in D.

We assume the existence of a partial order on $\mathcal{X}$ and a total order on $\mathcal{Y}$. Typically, the partial order on $\mathcal{X}$ is the product order induced by total orders on $\mathcal{X}_i$, that is

$$\mathbf{x} \leq \mathbf{x}' \Leftrightarrow x_i \leq x'_i \qquad \forall i = 1, \ldots, p. \tag{1}$$

The objective is to learn from data an allocation rule $c : \mathcal{X} \rightarrow \mathcal{Y}$ such that $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

$$\mathbf{x} \leq \mathbf{x}' \Rightarrow c(\mathbf{x}) \leq c(\mathbf{x}'), \tag{2}$$

that is, a lower ordered input is not allowed to have a higher class label. In case of the product order defined in (1) this constraint expresses the knowledge that each attribute has a positive influence on the class label.

It is customary to evaluate a classifier on the basis of its error-rate or 0/1 loss. For classification problems with ordered class labels this choice is less obvious. It makes sense to incur a higher cost for those misclassifications that are "far" from the true label, than to those that are "close". One loss function that has this property is L_1 loss:

$$L(i, j) = |i - j| \qquad i, j = 1, \ldots, k \tag{3}$$

where i is the true label, and j the predicted label. We note that this is certainly not the only possible choice. One could also choose L_2 loss for example, or another loss function that has the desired property that misclassifications that are far from the true label incur a higher loss. Nevertheless, L_1 loss is a reasonable candidate, and in this paper we confine our attention to this loss function.

3 The isotonic regression

In this section we give a short description of the isotonic regression. In the next section we discuss its application to monotone classification in MOCA.

Let $Z = \{z_1, z_2, \ldots, z_n\}$ be a nonempty finite set of constants and let $\leq$ be a partial order on Z. Any real-valued function f on Z is *isotonic* with respect to $\leq$ if, for any $z, z' \in Z$, $z \leq z'$ implies $f(z) \leq f(z')$. We assume that each element z_i of Z is associated with a real number $g(z_i)$; these real numbers typically are estimates of the function values of an unknown isotonic function on Z. Furthermore, each element of Z has associated a positive weight $w(z_i)$ that typically indicates the precision of this estimate. An isotonic function g^* on Z now is an *isotonic regression* of g with respect to the weight function w and the partial order $\leq$ if and only if it minimizes the sum

$$\sum_{i=1}^{n} w(z_i)\ [f(z_i) - g(z_i)]^2 \tag{4}$$

in the class of isotonic functions f on Z. Brunk [4] proved the existence of a unique g^*.

Any real-valued function f on Z is *antitonic* with respect to $\leq$ if, for any $z, z' \in Z$, $z \leq z'$ implies $f(z) \geq f(z')$. The antitonic regression of g is defined completely analogous to the isotonic regression as the function that minimizes (4) within the class of antitonic functions. The *isotonic* regression with respect to a partial order, is equivalent to the *antitonic* regression with respect to the inverse order.

The best time complexity known for an exact solution to the isotonic regression problem for arbitrary partial order is $O(n^4)$ [11]. It is based on a divide-and-conquer strategy that involves solving at most n maximal flow problems.

A subset L of Z is a *lower set* of Z with respect to $\leq$, if $z \in L$, $z' \in Z$, and $z' \leq z$ imply $z' \in L$. Hence, if a lower set contains a particular element, it is required to also contain all lower ordered elements. Likewise, a subset U of Z is an *upper set* of Z if $z \in U$, $z' \in Z$, and $z \leq z'$ imply $z' \in U$. The *weighted average* of g, with weights w, for a nonempty subset A of Z is defined as

$$\text{Av}(A, g) = \frac{\sum_{z \in A} w(z) g(z)}{\sum_{z \in A} w(z)} \tag{5}$$

A *maximal partition* of $Z = \{z_1, z_2, \ldots, z_n\}$ with respect to the isotonic regression is a partition $B_1, \ldots, B_m$ of nonempty sets such that

1. $g^*(z_j) = \text{Av}(B_i, g) \qquad \forall z_j \in B_i$
2. Each B_i can be written as the intersection of an upper and lower set, and
3. m is as large as possible.

The maximal partition can be computed by choosing each new lower set to have minimal cardinality in the Minimum Lower Sets algorithm [6]. Finally, we define the downset of z_0 with respect to Z to be the set $\{z \in Z : z \leq z_0\}$. The upset of z_0 is defined analogously.

4 MOCA

In this section, we describe a new nonparametric classification algorithm called MOCA. The objective of MOCA is to produce a classifier that satisfies (2), and subject to this constraint minimizes the mean absolute prediction error. MOCA can be regarded as a probabilistic classifier, in the sense that for each input vector observed in the training sample, it estimates the class distribution. Estimates of class distributions for other input vectors are obtained by interpolation. The MOCA estimates of the class distributions satisfy the *stochastic order constraint*:

$$\mathbf{x} \leq \mathbf{x}' \Rightarrow \tilde{F}_i(\mathbf{x}) \geq \tilde{F}_i(\mathbf{x}') \qquad i = 1, \ldots, k \tag{6}$$

where $\tilde{F}(\mathbf{x})$ denotes the MOCA estimate of the cumulative class probability distribution for input vector $\mathbf{x}$.

To get an outright class assignment, we take the smallest median of $\tilde{F}(\mathbf{x})$. Since $\tilde{F}(\mathbf{x})$ satisfies the stochastic order constraint (6), allocation to the median is guaranteed to satisfy the monotonicity property stated in (2). We show that the given allocation rule minimizes L_1 loss on the training sample subject to the monotonicity requirement. Although this result is not immediately obvious, it does seem plausible, since the median is known to minimize L_1 loss.

After this general description, we proceed with the technical details. Recall that Z is the set of distinct $\mathbf{x}$ values present in the training sample D. Let

$$\hat{P}_j(\mathbf{x}) = \frac{n(\mathbf{x}, j)}{n(\mathbf{x})}, \qquad \mathbf{x} \in Z$$

where $n(\mathbf{x})$ denotes the number of observations in D with attribute values $\mathbf{x}$, and $n(\mathbf{x}, j)$ denotes the number of observations in D with attribute values $\mathbf{x}$ and class label j. Furthermore, let

$$\hat{F}_i(\mathbf{x}) = \sum_{j \leq i} \hat{P}_j(\mathbf{x}), \qquad \mathbf{x} \in Z$$

denote the unconstrained maximum likelihood estimate of

$$F_i(\mathbf{x}) = P(y \leq i|\mathbf{x}), \mathbf{x} \in Z.$$

Definition 1 (MOCA estimator) *The* MOCA *estimator*

$$F_i^*(\mathbf{x}), \qquad i = 1, 2, \ldots, k; \quad \mathbf{x} \in Z$$

of $F_i(\mathbf{x})$ is given by the antitonic regression of $g(\mathbf{x}) = \hat{F}_i(\mathbf{x})$ with weights $w(\mathbf{x}) = n(\mathbf{x})$, for each value $i = 1, 2, \ldots, k$.

Note that this estimator satisfies the stochastic order constraint $\forall \mathbf{x}, \mathbf{x}' \in Z$:

$$\mathbf{x} \leq \mathbf{x}' \Rightarrow F_i^*(\mathbf{x}) \geq F_i^*(\mathbf{x}') \qquad i = 1, \ldots, k \quad (7)$$

by construction. It is therefore not surprising that it has been used for estimation under a stochastic order constraints in the past. It was proposed for linear orders already by Hogg [9], and later analyzed by El Barmi and Mukerjee [7]. It was used by Feelders [8] for parameter estimation in Bayesian networks under a stochastic order constraint.

Now the isotonic regression is only defined on the observed data points, that is, only for $\mathbf{x} \in Z$. Typically our training sample does not cover the entire input space, i.e. $Z \subset \mathcal{X}$, so we need some way to estimate $F_i(\mathbf{x}_0)$ for points $\mathbf{x}_0$ not in the training sample. Of course these estimates should satisfy the stochastic order constraint with respect to $F^*(\mathbf{x})$. Hence, we can derive the following bounds:

$$F_i^{\min}(\mathbf{x}_0) = \max_{\mathbf{x}_0 \leq \mathbf{x}} F_i^*(\mathbf{x}) \qquad i = 1, \ldots, k \quad (8)$$

and

$$F_i^{\max}(\mathbf{x}_0) = \min_{\mathbf{x} \leq \mathbf{x}_0} F_i^*(\mathbf{x}) \qquad i = 1, \ldots, k \quad (9)$$

If there is no point $\mathbf{x}$ in Z such that $\mathbf{x} \leq \mathbf{x}_0$, then we put $F_i^{\min}(\mathbf{x}_0) = 1$ $(i = 1, \ldots, k)$, and if there is no point $\mathbf{x}$ in Z such that $\mathbf{x}_0 \leq \mathbf{x}$, then we put $F_i^{\max}(\mathbf{x}_0) = 0$ $(i = 1, \ldots, k-1)$, and $F_k^{\max}(\mathbf{x}_0) = 1$.

Because F_i^* is antitonic we always have $F_i^{\min} \leq F_i^{\max}$. Any choice from the interval $[F_i^{\min}, F_i^{\max}]$ satisfies the stochastic order constraint with respect to the training data.

A simple interpolation scheme that is guaranteed to produce globally consistent estimates is to take the convex combination

$$\tilde{F}_i(\mathbf{x}_0) = \alpha F_i^{\min}(\mathbf{x}_0) + (1 - \alpha) F_i^{\max}(\mathbf{x}_0),$$

with $\alpha \in [0, 1]$. Note that for $\mathbf{x}_0 \in Z$, we have $\tilde{F}_i(\mathbf{x}_0) = F_i^*(\mathbf{x}_0)$, since both $F_i^{\min}(\mathbf{x}_0)$ and $F_i^{\max}(\mathbf{x}_0)$ are equal to $F_i^*(\mathbf{x}_0)$. The value of α can be chosen so as to minimize empirical loss on a test sample.

Since MOCA should produce a class prediction, we still have to specify an allocation rule. MOCA allocates $\mathbf{x}$ to the smallest median of $\tilde{F}(\mathbf{x})$:

$$c_{\text{MOCA}}(\mathbf{x}) = \min_i : \tilde{F}_i(\mathbf{x}) \geq 0.5$$

First of all, note that since $\tilde{F}_i$ satisfies the stochastic order constraint (6), c_{MOCA} will satisfy the monotonicity constraint given in (2). Furthermore, it can be shown that c_{MOCA} minimizes L_1 loss

$$\sum_{i=1}^{N} |y_i - c(\mathbf{x}_i)|$$

within the class of monotone integer-valued functions $c(\cdot)$. In other words, of all monotone classifiers, c_{MOCA} is among the ones (there may be more than one) that minimize L_1 loss on the training sample. We prove this as follows: Dykstra et al. [6] describe a method for minimizing L_1 loss that they prove correct. We show that c_{MOCA} satisfies all the requirements of their method.

In [6] Dykstra et al. compute the isotonic regression $p_i^*(\mathbf{x})$ of $p_i(\mathbf{x}) = 1 - \hat{F}_i(\mathbf{x})$, with weights $w(\mathbf{x}) = n(\mathbf{x})$. It is not difficult to show that

$$p_i^*(\mathbf{x}) = 1 - F_i^*(\mathbf{x}) \qquad i = 1, \ldots, k-1 \quad (10)$$

Next, Dykstra et al. [6] show that an allocation rule $c(\mathbf{x})$ minimizes L_1 loss on the training sample if it satisfies three properties. Using (10) we can write these properties as:

1. If $F_i^*(\mathbf{x}) > \frac{1}{2}$ then $c(\mathbf{x}) \leq i$, for $i = 1, \ldots, k-1$.
2. If $F_i^*(\mathbf{x}) < \frac{1}{2}$ then $c(\mathbf{x}) > i$, for $i = 1, \ldots, k-1$.
3. $c(\mathbf{x})$ is constant (either i or $i+1$) on every element of the maximal partition that is a subset of $\{\mathbf{x} : F_i^*(\mathbf{x}) = \frac{1}{2}\}$, for $i = 1, \ldots, k-1$.

We show that $c_{\text{MOCA}}(\mathbf{x})$ has the desired properties.

1. It follows from the definition of c_{MOCA} that if $F_i^*(\mathbf{x}) > \frac{1}{2}$ then $c_{\text{MOCA}}(\mathbf{x}) \leq i$.
2. Likewise, it follows from the definition of c_{MOCA} that if $F_i^*(\mathbf{x}) < \frac{1}{2}$ then $c_{\text{MOCA}}(\mathbf{x}) > i$.
3. $c_{\text{MOCA}}(\mathbf{x}) = i$ on $\{\mathbf{x} : F_i^*(\mathbf{x}) = \frac{1}{2}\}$, for $i = 1, \ldots, k-1$.

The first two conditions are straightforward, but the third one is a bit more involved; therefore we illustrate it with an example. Suppose we have a data set $D = \{(1, 1, 3), (1, 2, 1), (2, 1, 1), (2, 1, 3)\}$, where each tuple has the form (x_1, x_2, y). Table 1 contains these 4 observations on 3 distinct input vectors, together with the ML and MOCA estimates of $F_i(\mathbf{x})$. Note that $\hat{F}$ violates the order constraints, and the antitonic regression removes this violation by averaging $\hat{F}_i(\mathbf{x})$ over the cells $(1, 1)$ and $(1, 2)$, for $i = 1$

	y			$\hat{F}$		F^*	
(x_1, x_2)	1	2	3	1	2	1	2
$(1,1)$	0	0	1	0	0	1/2	1/2
$(1,2)$	1	0	0	1	1	1/2	1/2
$(2,1)$	1	0	1	1/2	1/2	1/2	1/2

Table 1. Data (left), maximum likelihood (middle) and MOCA estimates (right) for example.

as well as $i = 2$. This results in F^* as given in the rightmost part of the table. Note also that the maximal partition is given by $B_1 = \{(2,1)\}$ and $B_2 = \{(1,1),(1,2)\}$, both for $i = 1$ and $i = 2$. The set of medians of F^* for all three input vectors is $\{1,2,3\}$. The third condition of Dykstra et al. states that one can assign any of these three values (as long as the assignment satisfies the monotonicity constraint), but one should assign the same labels to the elements $(1,1)$ and $(1,2)$ of B_2, because they were averaged in computing F^*. The reader can easily verify that assigning different labels to them leads to suboptimal L_1 error. We can assign any label to $(2,1)$ however, as long as it is consistent with the label we assigned to the other two input vectors. For example, if we put $c(1,1) = c(1,2) = 2$, then we can still assign either 2 or 3 to $(2,1)$: both give the minimum possible L_1 error. Now MOCA will assign the label 1 to all three input vectors, and hence it is constant on a larger set of input vectors than required. This does not harm the optimality of the assignment however.

5 Related work

The proposed algorithm is closely related to the ordinal stochastic dominance learner (OSDL) developed by Cao-Van [5] and generalized by Lievens et al. in [10]. We give a short description of OSDL to point out the similarities and differences with MOCA.

To obtain a collection of distribution functions that satisfy the stochastic order requirement, Cao-Van [5] defines:

$$F_i^{\min}(\mathbf{x}_0) = \min_{\mathbf{x} \leq \mathbf{x}_0} \hat{F}_i(\mathbf{x}) \tag{11}$$

and

$$F_i^{\max}(\mathbf{x}_0) = \max_{\mathbf{x}_0 \leq \mathbf{x}} \hat{F}_i(\mathbf{x}), \tag{12}$$

where $\mathbf{x} \in Z$. Note that $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

$$\mathbf{x} \leq \mathbf{x}' \quad \Rightarrow F_i^{\min}(\mathbf{x}) \geq F_i^{\min}(\mathbf{x}') \tag{13}$$

$$\mathbf{x} \leq \mathbf{x}' \quad \Rightarrow F_i^{\max}(\mathbf{x}) \geq F_i^{\max}(\mathbf{x}'). \tag{14}$$

Proposition (13) holds, since the downset of $\mathbf{x}$ is a subset of the downset of $\mathbf{x}'$, and the minimum taken over a given set is never above the minimum taken over one of its subsets. Proposition (14) follows similarly.

The final estimates are obtained by putting

$$\tilde{F}_i(\mathbf{x}) = \alpha F_i^{\min}(\mathbf{x}_0) + (1-\alpha) F_i^{\max}(\mathbf{x}_0), \tag{15}$$

with $\alpha \in [0,1]$.

This rule is used both for observed data points, as well as for new data points. Like with MOCA, α is a free parameter whose value can be selected so as to minimize empirical loss on a test sample. Note that $\tilde{F}$ satisfies the stochastic order constraint, because both (13) and (14) hold.

The reader will have noticed the similarity between MOCA and OSDL: MOCA uses the same interpolation method, and the MOCA definitions of $F^{\min}$ and $F^{\max}$ are the reverse of the corresponding definitions for OSDL. The important difference is that OSDL plugs in the maximum likelihood estimates $\hat{F}$ in equations (11) and (12), whereas MOCA plugs in the isotonic regression estimates F^* in equations (8) and (9). The most important consequence of this difference is that MOCA is guaranteed to minimize L_1 loss on the training sample, whereas this is not the case for OSDL. Another difference is the choice of allocation rule. Originally Cao-Van [5] assigned $\mathbf{x}$ to the expected value of $\tilde{F}(\mathbf{x})$, rounded to the nearest integer. In [10] the allocation rule is changed to a median of $\tilde{F}(\mathbf{x})$, but the choice of median is left unspecified, provided that it is chosen in such a way that the monotonicity constraint is satisfied.

6 Example

In this section we present a small example to illustrate both MOCA and OSDL. Suppose we have a problem with two input attributes X_1 and X_2, and a class label Y, all of them taking their values from the set $\{1,2,3\}$. Hence we have

$$\mathcal{X} = \{1,2,3\} \times \{1,2,3\} \quad \text{and} \quad \mathcal{Y} = \{1,2,3\}.$$

The observed data and the maximum likelihood estimates $\hat{F}$ are given in table 2. Note that

$$Z = \{(1,1),(1,2),(2,1),(1,3),(3,2)\}$$

in this example.

The data point $(3,2)$ with class label 1 "spoils" the monotonicity of $\hat{F}$. In table 3 we give the MOCA and OSDL estimates of F on the observed attribute vectors, together with their median values. For OSDL we used $\alpha = \frac{1}{2}$; for MOCA the value of α is immaterial since the estimate will always be equal to F^* on the observed attribute vectors. MOCA resolves the violation by taking the weighted average of $\hat{F}_1(2,1)$ and $\hat{F}_1(3,2)$ and assigning this value to both cells

$$F_1^*(2,1) = F_1^*(3,2) = \frac{3 \times 0 + 1 \times 1}{3+1} = \frac{1}{4}$$

		y				$\hat{F}$	
	(x_1, x_2)	1	2	3	n	1	2
1	$(1,1)$	2	0	0	2	1	1
2	$(1,2)$	1	2	0	3	1/3	1
3	$(2,1)$	0	2	1	3	0	2/3
4	$(1,3)$	0	0	1	1	0	0
5	$(3,2)$	1	0	0	1	1	1

Table 2. Data and ML estimates for example.

	MOCA			OSDL		
(x_1, x_2)	1	2	Med.	1	2	Med.
$(1,1)$	1	1	1	1	1	1
$(1,2)$	1/3	1	2	2/3	1	1
$(2,1)$	1/4	3/4	2	1/2	5/6	{1,2}
$(1,3)$	0	0	3	0	0	3
$(3,2)$	1/4	3/4	2	1/2	5/6	{1,2}

Table 3. MOCA **and** OSDL **estimates, and the corresponding medians.**

The order violation in $\hat{F}_2$ is dealt with in a similar manner.

To illustrate OSDL, we show how $\tilde{F}_1(1,2)$ is computed. We have

$$\begin{aligned} F_1^{\min}(1,2) &= \min\{1/3, 1\} = 1/3 \\ F_1^{\max}(1,2) &= \max\{1/3, 0, 1\} = 1 \\ \tilde{F}_1(1,2) &= \tfrac{1}{2} \times \tfrac{1}{3} + \tfrac{1}{2} \times 1 = \tfrac{2}{3} \end{aligned}$$

The L_1 error of c_{MOCA} is given by

$$L_1[c_{\text{MOCA}}] = 0 + 1 + 1 + 0 + 1 = 3.$$

This is the minimum possible L_1 error on the training data for a monotone classifier. For c_{OSDL} we have a choice of medians for the third and fifth observation. The lowest error is obtained if we assign both the label 2:

$$L_1[c_{\text{OSDL}}] = 0 + 2 + 1 + 0 + 1 = 4$$

7 Experiments

We performed experiments on a number of data sets in order to compare our method to OSDL with respect to their average L_1 errors. We performed experiments on both artificial and real datasets. These are discussed separately in section 7.1 and section 7.2. In all experiments we have assumed that, like MOCA, OSDL assigns $\mathbf{x}$ to the smallest median of $\tilde{F}(\mathbf{x})$.

σ^2	L_1^{MOCA}	L_1^{OSDL}	L_1^{MOCA}	L_1^{OSDL}
0	0.3009	0.3009	0.621	0.6549
$(1/10)M$	0.3359	0.3688	0.7297	0.8974
$(2/10)M$	0.5039	0.5441	0.811	0.9991
$(3/10)M$	0.6472	0.7096	0.8727	1.2763
$(4/10)M$	0.7736	0.8641	1.0004	1.3331
$(5/10)M$	0.8453	0.9616	1.0207	1.4066
$(6/10)M$	0.9623	1.1389	1.0964	1.4556
$(7/10)M$	0.9297	1.2999	1.0463	1.3733
$(8/10)M$	0.9762	1.3428	1.0245	1.4614
$(9/10)M$	1.0263	1.3558	1.0944	1.4259
M	1.0195	1.4251	1.0248	1.4583

Table 4. Experimental results on the artificial data generated by the monotone function f_1 (left) and the non-monotone function f_2 (right).

7.1 Artificial Data

To compare the performance of MOCA and OSDL in controlled circumstances we generated artificial data from a monotone function f_1,

$$f_1(x_1, x_2) = 1 + x_1 + \tfrac{1}{2}(x_2^2 - x_1^2) \qquad (16)$$

and from a non-monotone function f_2,

$$f_2(x_1, x_2) = 3 + \sin\left(\frac{\pi}{2} x_1\right)(2 + \sin(2\pi x_2)) \qquad (17)$$

where x_1 and x_2 are drawn independently from the uniform distribution on the unit interval. The non-monotone function f_2 was used to test the robustness of the algorithms against violation of the monotonicity assumption.

We sampled 100 points for training, and another 10,000 to get a reliable estimate of the mean absolute prediction error. Then we added a normally distributed error term with mean zero and variance σ^2 to each value of f_1 and f_2. To create ordered class labels, the resulting numeric values were discretized into four intervals in such a way that each contained approximately the same number of cases. We tried values for $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and picked the best value (i.e. the one with the lowest error on the test set) for the final comparison between MOCA and OSDL.

To study the behaviour at different levels of noise, we tried $\sigma^2 \in \{\frac{k}{10}M\}_{k=0}^{10}$, where M is the maximum observed value of equation (16) and equation (17) respectively. Note that even though f_1 is a monotone function, the data may contain non-monotone pairs of observations, due to the addition of noise. The non-monotone function f_2 will contain non-monotone pairs even at the zero noise level.

The results are given in table 4. We observe that MOCA has consistently lower error, except of course for the monotone data without noise: in that case $\hat{F}$ already satisfies the stochastic order constraint, and hence MOCA and OSDL give identical results. All observed improvements are significant at $\alpha = 0.01$.

7.2 Real data

For the experiments on real data, we selected a number of data sets for which the presence of an increasing (or decreasing) relation between the attributes and the response variable was plausible. They are available from the UCI machine learning repository [2] except for *Windsor Housing*[1] [1] and *Employee Selection* [2] [3].

As for the Australian credit approval data, we only used columns 7, 8, 9 and 10 of the attributes from the original data set. For the Boston housing data, we excluded the *Charles River* dummy variable. Several of the data sets we used had a binary target variable, one had a 9-class target variable (the Employee Selection data set), one had a 4-class target variable (the Car data set) and the remaining data sets had a numeric target. The numeric targets were discretized into four intervals, in such a way that each interval contained approximately the same number of observations.

For each data set, we selected the best α value from the set $\{0, 0.25, 0.5, 0.75, 1\}$, both for MOCA and OSDL, using 10-fold cross-validation. We then picked the best result obtained for each method in terms of the average L_1 error and compared them by performing a paired sample t-test. The results are given in Table 5. We note that MOCA has lower error in 7 out of 9 cases, in 3 cases significant at $\alpha = 0.05$. In two cases OSDL is better, and significantly so on the Car data set.

Data set (# classes)	L_1^{MOCA}	L_1^{OSDL}
Australian credit (2)	$\mathbf{0.161}^*$	0.336
AutoMpg (4)	**0.253**	0.255
Boston housing (4)	**0.457**	0.502
Car (4)	0.041	$\mathbf{0.032}^*$
ESL (9)	**0.334**	0.344
Haberman survival (2)	0.261	**0.258**
Machine (4)	$\mathbf{0.340}^*$	0.383
Pima indians (2)	**0.260**	0.266
Windsor housing (4)	$\mathbf{0.538}^*$	0.593

Table 5. Experimental results on the real data sets. Lower error is shown in boldface. $*$ indicates a significant difference at $\alpha = 0.05$.Note that for the binary classification problems, the reported error is equal to the error-rate.

8 Conclusion

We have presented MOCA, a new nonparametric monotone classification algorithm that attempts to minimize the mean absolute prediction error for classification problems with ordered class labels. We have shown that MOCA minimizes the L_1 error on the training sample, subject to the monotonicity constraint. Through experiments on artificial and real data, we have shown that it compares favourably to OSDL, a classification algorithm intended for the same type of monotone classification problems as MOCA.

[1] Available from the Journal of Applied Econometrics Data Archive at `http://econ.queensu.ca/jae/`

[2] Available at `http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html`

References

[1] P. Anglin and R. Gençay. Semiparametric estimation of a hedonic price function. *Journal of Applied Econometrics*, 11(6):633–648, 1996.

[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[3] A. Ben-David, L. Sterling, and Y. Pao. Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, 5:45–49, 1989.

[4] H. Brunk. Conditional expectation given a σ-lattice and applications. *Annals of Mathematical Statistics*, 36:1339–1350, 1965.

[5] K. Cao-Van. *Supervised ranking, from semantics to algorithms*. PhD thesis, Universiteit Gent, 2003.

[6] R. Dykstra, J. Hewett, and T. Robertson. Nonparametric, isotonic discriminant procedures. *Biometrika*, 86(2):429–438, 1999.

[7] H. El Barmi and H. Mukerjee. Inferences under a stochastic ordering constraint: the k-sample case. *Journal of the American Statistical Association*, 100(469):252–261, 2005.

[8] A. Feelders. A new parameter learning method for Bayesian networks with qualitative influences. In R. Parr and L. v. d. Gaag, editors, *Proceedings of Uncertainty in Artificial Intelligence 2007 (UAI07)*, pages 117–124. AUAI Press, 2007.

[9] R. Hogg. On models and hypotheses with restricted alternatives. *Journal of the American Statistical Association*, 60(312):1153–1162, 1965.

[10] S. Lievens, B. De Baets, and K. Cao-Van. A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting. *Annals of Operations Research*, DOI 10.1007/s10479-008-0326-1, 2008.

[11] W. Maxwell and J. Muckstadt. Establishing consistent and realistic reorder intervals in production-distribution systems. *Operations Research*, 33(6):1316–1341, 1985.

[12] T. Robertson, F. Wright, and R. Dykstra. *Order Restricted Statistical Inference*. Wiley, 1988.